In [1]:

```
# importing various libraries which are used
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import preprocessing as p
from scipy.stats import f
from sklearn.metrics import mean_squared_error, mean_absolute_error ,r2_score
```

In [2]:

```
# taking the csv file as input to create the model
file=input("csv file_name : ")
```

```
csv file_name : 507.csv
```

In [3]:

```
# creating a pandas dataframe using the values obtained in the csv
df=pd.read_csv(file)
```

In [4]:

```
# printing the head of the data frame to get the gist of the values
print("\n data frame head :- \n",df.head())
```

```
 data frame head :-
        time  cpu-cycles  instructions  l1d.replacement  icache_64b.if
tag_miss  \
0  0.100148   457605823    1040202613         12468216
1102989
1  0.200431   459564526     821360749         40433713
286218
2  0.300652   458946377     812830085         58288187
24785
3  0.400888   459383035     864041757         67366094
303846
4  0.501103   459288454     809577517         60247007
312258

   l2_rqsts.all_demand_miss  longest_lat_cache.miss  \
0                   1639598                 1691351
1                   2042036                 3006192
2                   1403075                 3413324
3                   1287810                 1013444
4                   1920982                 2424261

   br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                      46486343                        2284
1                      47075138                         109
2                      50380490                         107
3                       8991915                           1
4                      27355751                        7216

   itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                        7370                            13548
1                         314                            20104
2                         286                            75252
3                          14                            18165
4                         412                            15812

   dtlb_store_misses.walk_completed  branch-misses
0                            209744          30207
1                            396182          47991
2                             24377          43049
3                             49650           7031
4                            119746           9544
```

In [5]:

```
# creating a new col in our data frame which consists of the CPI per tuple
df[['CPI']]=df[['cpu-cycles']].div(df['instructions'], axis=0)
print(df)
```

```
            time  cpu-cycles  instructions  l1d.replacement  \
0       0.100148   457605823    1040202613         12468216
1       0.200431   459564526     821360749         40433713
2       0.300652   458946377     812830085         58288187
3       0.400888   459383035     864041757         67366094
4       0.501103   459288454     809577517         60247007
...          ...         ...           ...              ...
1468  147.238183   459170843     735244397        115580692
1469  147.338451   459394661     831808139         57997353
1470  147.438670   458863493     880420773         69512796
1471  147.538882   459280685     643031046         58578149
1472  147.571908   151041064     476208708          3909102

      icache_64b.iftag_miss  l2_rqsts.all_demand_miss  longest_lat_cac
he.miss  \
0                   1102989                   1639598
1691351
1                    286218                   2042036
3006192
2                     24785                   1403075
3413324
3                    303846                   1287810
1013444
4                    312258                   1920982
2424261
...                     ...                       ...
...
1468                  19003                   2364298
7129173
1469                  14489                    934872
3402560
1470                  58314                    482565
807239
1471                  21025                    757232
1872916
1472                 188452                    711112
1134867

      br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                         46486343                        2284
1                         47075138                         109
2                         50380490                         107
3                          8991915                           1
4                         27355751                        7216
...                            ...                         ...
1468                      16837765                          23
1469                      46237001                         136
1470                       5351185                           5
1471                      16258834                          66
1472                      82741184                        2284

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                           7370                            13548
1                            314                            20104
2                            286                            75252
3                             14                            18165
4                            412                            15812
...                          ...                              ...
1468                          64                            80662
1469                         285                            92811
1470                          34                            13651
```

```
1471                            55                                6981
1472                          4299                                5034

      dtlb_store_misses.walk_completed  branch-misses       CPI
0                               209744          30207  0.439920
1                               396182          47991  0.559516
2                                24377          43049  0.564628
3                                49650           7031  0.531668
4                               119746           9544  0.567319
...                                ...            ...       ...
1468                               530           3294  0.624515
1469                             32130          32554  0.552284
1470                              6042           3834  0.521187
1471                             33918           9362  0.714243
1472                              6449         115434  0.317174

[1473 rows x 14 columns]
```

In [6]:

```
# dividing all the values by instruction so that we get values in each coloumn per i
df[['l1d.replacement','icache_64b.iftag_miss','l2_rqsts.all_demand_miss','longest_la
print(df)
```

```
            time  cpu-cycles  instructions  l1d.replacement  \
0       0.100148   457605823    1040202613         0.011986
1       0.200431   459564526     821360749         0.049228
2       0.300652   458946377     812830085         0.071710
3       0.400888   459383035     864041757         0.077966
4       0.501103   459288454     809577517         0.074418
...          ...         ...           ...              ...
1468  147.238183   459170843     735244397         0.157200
1469  147.338451   459394661     831808139         0.069724
1470  147.438670   458863493     880420773         0.078954
1471  147.538882   459280685     643031046         0.091097
1472  147.571908   151041064     476208708         0.008209

      icache_64b.iftag_miss  l2_rqsts.all_demand_miss  longest_lat_cac
he.miss  \
0                  0.001060                  0.001576
0.001626
1                  0.000348                  0.002486
0.003660
2                  0.000030                  0.001726
0.004199
3                  0.000352                  0.001490
0.001173
4                  0.000386                  0.002373
0.002994
...                     ...                       ...
...
1468               0.000026                  0.003216
0.009696
1469               0.000017                  0.001124
0.004091
1470               0.000066                  0.000548
0.000917
1471               0.000033                  0.001178
0.002913
1472               0.000396                  0.001493
0.002383

      br_inst_retired.all_branches  frontend_retired.itlb_miss  \
0                         0.044690                2.195726e-06
1                         0.057314                1.327066e-07
2                         0.061982                1.316388e-07
3                         0.010407                1.157351e-09
4                         0.033790                8.913291e-06
...                            ...                         ...
1468                      0.022901                3.128212e-08
1469                      0.055586                1.634992e-07
1470                      0.006078                5.679103e-09
1471                      0.025285                1.026389e-07
1472                      0.173750                4.796216e-06

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                   7.085158e-06                         0.000013
1                   3.822924e-07                         0.000024
2                   3.518571e-07                         0.000093
3                   1.620292e-08                         0.000021
4                   5.089074e-07                         0.000020
...                          ...                              ...
1468                8.704589e-08                         0.000110
1469                3.426271e-07                         0.000112
1470                3.861790e-08                         0.000016
```

```
1471                 8.553242e-08                          0.000011
1472                 9.027554e-06                          0.000011

      dtlb_store_misses.walk_completed  branch-misses       CPI
0                           2.016376e-04       0.000029  0.439920
1                           4.823483e-04       0.000058  0.559516
2                           2.999028e-05       0.000053  0.564628
3                           5.746250e-05       0.000008  0.531668
4                           1.479117e-04       0.000012  0.567319
...                                  ...            ...       ...
1468                        7.208487e-07       0.000004  0.624515
1469                        3.862670e-05       0.000039  0.552284
1470                        6.862628e-06       0.000004  0.521187
1471                        5.274706e-05       0.000015  0.714243
1472                        1.354238e-05       0.000242  0.317174

[1473 rows x 14 columns]
```

In [7]:

```
# droping values such as time , instructions , cpu-cycles and br_inst_retired.all_br
df= df.drop(['time'], axis=1)
df= df.drop(['instructions'], axis=1)
df= df.drop(['cpu-cycles'], axis=1)
df= df.drop(['br_inst_retired.all_branches'], axis=1)
```

In [8]:

```
# assigning y as the CPI and then droping it from the dataframe
y=df['CPI']
df= df.drop(['CPI'], axis=1)
print("y values :- \n",y)
```

```
y values :-
 0       0.439920
1       0.559516
2       0.564628
3       0.531668
4       0.567319
          ...
1468    0.624515
1469    0.552284
1470    0.521187
1471    0.714243
1472    0.317174
Name: CPI, Length: 1473, dtype: float64
```

In [9]:

```
# assigning x as the dataframe
x=df
print("x values :- \n",x)
```

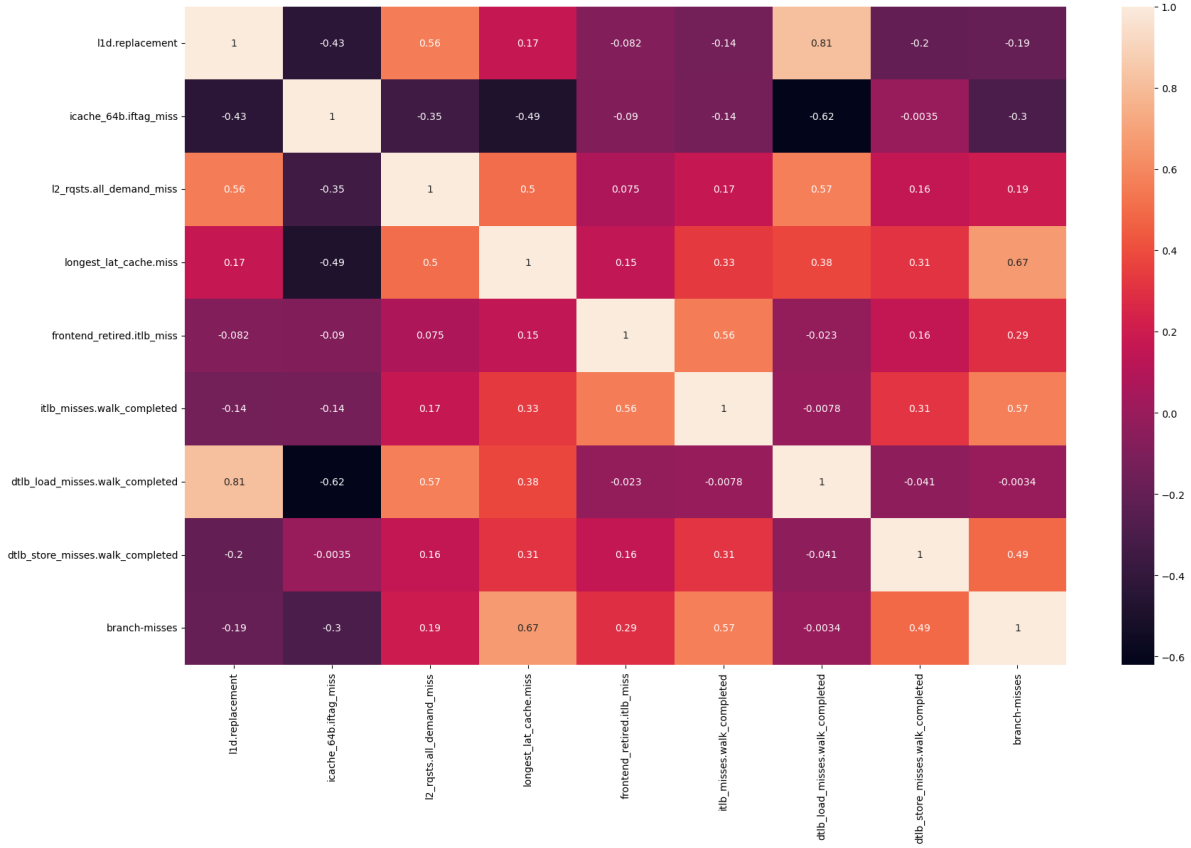
```
x values :-
       l1d.replacement  icache_64b.iftag_miss  l2_rqsts.all_demand_mis
s  \
0             0.011986               0.001060                 0.001576
1             0.049228               0.000348                 0.002486
2             0.071710               0.000030                 0.001726
3             0.077966               0.000352                 0.001490
4             0.074418               0.000386                 0.002373
...                ...                    ...                      ...
1468          0.157200               0.000026                 0.003216
1469          0.069724               0.000017                 0.001124
1470          0.078954               0.000066                 0.000548
1471          0.091097               0.000033                 0.001178
1472          0.008209               0.000396                 0.001493

      longest_lat_cache.miss  frontend_retired.itlb_miss  \
0                   0.001626                2.195726e-06
1                   0.003660                1.327066e-07
2                   0.004199                1.316388e-07
3                   0.001173                1.157351e-09
4                   0.002994                8.913291e-06
...                      ...                         ...
1468                0.009696                3.128212e-08
1469                0.004091                1.634992e-07
1470                0.000917                5.679103e-09
1471                0.002913                1.026389e-07
1472                0.002383                4.796216e-06

      itlb_misses.walk_completed  dtlb_load_misses.walk_completed  \
0                   7.085158e-06                         0.000013
1                   3.822924e-07                         0.000024
2                   3.518571e-07                         0.000093
3                   1.620292e-08                         0.000021
4                   5.089074e-07                         0.000020
...                          ...                              ...
1468                8.704589e-08                         0.000110
1469                3.426271e-07                         0.000112
1470                3.861790e-08                         0.000016
1471                8.553242e-08                         0.000011
1472                9.027554e-06                         0.000011

      dtlb_store_misses.walk_completed  branch-misses
0                         2.016376e-04       0.000029
1                         4.823483e-04       0.000058
2                         2.999028e-05       0.000053
3                         5.746250e-05       0.000008
4                         1.479117e-04       0.000012
...                                ...            ...
1468                      7.208487e-07       0.000004
1469                      3.862670e-05       0.000039
1470                      6.862628e-06       0.000004
1471                      5.274706e-05       0.000015
1472                      1.354238e-05       0.000242

[1473 rows x 9 columns]
```

In [10]:

```
# creating a heatmap of the correlation matrix
fig,axis = plt.subplots(figsize = (20,12))
sns.heatmap(x.corr(),annot=True)
```

Out[10]:

```
<AxesSubplot:>
```



In [11]:

```
# dividing the data set into test and train set in a 20:80 ration with a random stat
# the model trains on a particular set of values on every execution
X_train, X_test, y_train, y_test = train_test_split(
    x, y, test_size=.20,random_state=55)
```

In [12]:

```
# using MinMax Scaler to scale the data within the given range of 0 to 1 such that
#shape of the original distribution is same after transformation
mms = p.MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.transform(X_test)
```

In [13]:

```
print("X_train :-\n",X_train)
```

```
X_train :-
 [[4.60596222e-01 8.22171426e-01 6.03112615e-02 ... 8.13150100e-02
  5.03322639e-03 1.14193015e-05]
 [4.60059944e-01 8.23786311e-01 5.55550854e-02 ... 8.06307281e-02
  5.02893526e-03 1.55663000e-04]
 [4.60873033e-01 8.16387169e-01 5.06644904e-02 ... 7.91788527e-02
  4.98994186e-03 3.88999321e-04]
 ...
 [8.33520270e-01 8.02869373e-04 2.15405819e-01 ... 8.75809095e-01
  2.31896140e-05 6.47330886e-03]
 [4.60767565e-01 8.25080531e-01 4.20532766e-02 ... 7.53810987e-02
  5.05438867e-03 4.97495700e-04]
 [3.43580034e-01 6.62316543e-04 4.49061432e-02 ... 6.20783585e-01
  1.63876844e-02 1.58592804e-01]]
```

In [14]:

```
# mean of all the columns of the training set
df2 = X_train.mean(axis=0)
print(df2)
```

```
[0.58713327 0.34102894 0.13612227 0.12614972 0.00775885 0.01536588
 0.38737428 0.01320913 0.05320172]
```

In [15]:

```
# creating a linear regression model using sklearn.linear_model
model = LinearRegression(positive=True)
model.fit(X_train,y_train)
```

Out[15]:

```
LinearRegression(positive=True)
```

In [16]:

```
# finding the coefficients given by our model
c=model.coef_
print("\nCoefficients :- \n",c)
```

```
Coefficients :-
 [0.17970843 0.04731276 0.08163528 0.2709634  0.         0.
 0.         0.2487429  0.42078214]
```

In [17]:

```
# model intercept i.e. the " Base CPI "
i=model.intercept_
print("\nBase CPI : ",i)
```

```
Base CPI :  0.40479795084978815
```

In [18]:

```
# making the predictions using our model on the test set
predictions = model.predict(X_test)
```

In [19]:

```
# Actual CPI
ACPI = y_test.mean()
print("\n Actual CPI : ",ACPI)
```

```
 Actual CPI :  0.5981124979170619
```

In [20]:

```
# Predicted CPI
PCPI = predictions.mean()
print("\n Predicted CPI : ",PCPI)
```

```
 Predicted CPI :  0.5951329067233786
```

In [21]:

```
# Finding out RMSE , R^2 , adjusted R^2 using our predictions and test set
MSE = mean_squared_error(y_test, predictions)
rint("\n RMSE : ",RMSE)

2 = r2_score(y_test, predictions)
rint("\n R^2 : ",r2_score(y_test, predictions))

djusted_r2 = 1 - ( 1-model.score(X_test,y_test) ) * ( len(y_test) - 1 ) / ( len(y_tes
rint("\n adjusted R^2 : ",adjusted_r2)
```

```
 RMSE :  0.002160477910632003

 R^2 :  0.7253023662817404

 adjusted R^2 :  0.7166277041643216
```

In [22]:

```
# finding absolute error and accuracy on test set
err = mean_absolute_error(y_test, predictions)
print ( "\n Test error is :" , err *100 , "% " )
print ( "\n Test Accuracy is :" , (1- err) *100 , "%" )
```

```
 Test error is : 2.3841944058289197 %

 Test Accuracy is : 97.61580559417108 %
```

In [23]:

```
# F-statistic value which should be > 2.5 and p-value which should be < 0.05
F = (R2/(1-R2))*((X_test.shape[0]-1-X_test.shape[1])/X_test.shape[1])
print("\n F-statistic : ",F)

p = 1-f.cdf(F,X_test.shape[1],(X_test.shape[0]-1-X_test.shape[1]))
print("\n p-value : ",p)
```

```
 F-statistic :  83.61159852270592

 p-value :  1.1102230246251565e-16
```

In [24]:

```
#no of coefficients
X_test.shape[1]
```

Out[24]:

```
9
```

In [25]:

```
# no of tuples in the test set
X_test.shape[0]
```

Out[25]:

```
295
```

In [26]:

```
# finding the residual for our test set
residuals = y_test - predictions
print("\n Residual :- \n ",residuals)
```

```
 Residual :-
  494     0.001052
236    -0.050497
622     0.001841
1175   -0.042972
27     -0.013495
          ...
108     0.001106
389     0.030249
839    -0.011184
924     0.002112
1262   -0.031400
Name: CPI, Length: 295, dtype: float64
```

In [27]:

```
# residual graph
data = {
    'predicted': [i for i in predictions],
    'residuals': [i for i in residuals]
}

dfr = pd.DataFrame(data)
sns.scatterplot(data=dfr, x="predicted", y="residuals")
```

Out[27]:

```
<AxesSubplot:xlabel='predicted', ylabel='residuals'>
```